

编程的乐趣

用 Python 解算法谜题

[美] 斯里尼·德瓦达斯 (Srini Devadas) ◎ 著
戴 旭 李亚舟 许亚运 ◎ 译

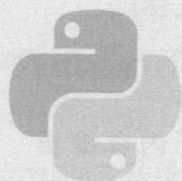
PROGRAMMING FOR THE PUZZLED:
LEARN TO PROGRAM WHILE SOLVING PUZZLES



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



编程的乐趣

用 Python 解算法谜题

[美] 斯里尼·德瓦达斯 (Srin Devadas) © 著

戴 旭 李亚舟 许亚运 © 译

人 民 邮 电 出 版 社
北 京

图书在版编目 (C I P) 数据

编程的乐趣：用Python解算法谜题 / (美) 斯里尼·德瓦达斯 (Srini Devadas) 著；戴旭，李亚舟，许亚运译. — 北京：人民邮电出版社，2019.6

(Programming for the Puzzled: Learn to Program While Solving Puzzles)
ISBN 978-7-115-50943-7

I. ①编… II. ①斯… ②戴… ③李… ④许… III. ①软件工具—程序设计 IV. ①TP311.561

中国版本图书馆CIP数据核字(2019)第043719号

内 容 提 要

这是一本介绍通过解决复杂谜题来学习编程的书，书中的代码用 Python 语言编写。与以往的编程书不同，本书将对代码功能的理解与编程语言语法和语义的理解分离开来，从解每个谜题开始，先给出解谜题的算法，随后用 Python 语法和语义实现对应的算法，并适当做出解释。本书包含了 21 个谜题，其中很多谜题都广为流传，如多皇后、汉诺塔、在几秒钟内解决数独问题、验证六度分隔猜想等，每个谜题后面都配有不同难度的编程习题，帮读者加深对相关算法的理解。

本书在算法谜题的趣味性和计算机编程的实用性之间搭建了一座桥梁，内容饶有趣味，讲述易于理解，适合已掌握初级编程概念并对算法感兴趣的学习者阅读和参考。

-
- ◆ 著 [美] 斯里尼·德瓦达斯 (Srini Devadas)
译 戴 旭 李亚舟 许亚运
责任编辑 杨海玲
责任印制 焦志炜
- ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
大厂聚鑫印刷有限责任公司印刷
- ◆ 开本：720×960 1/16
印张：15
字数：278 千字 2019 年 6 月第 1 版
印数：1—3 000 册 2019 年 6 月河北第 1 次印刷
- 著作权合同登记号 图字：01-2018-3261 号
-

定价：59.00 元

读者服务热线：(010)81055410 印装质量热线：(010)81055316

反盗版热线：(010)81055315

广告经营许可证：京东工商广登字 20170147 号

前言

谜题趣味非凡。顶级谜题的解可没那么浅显易得，需要灵光一闪才能发现。算法谜题是指谜题的解法就是算法，解题的步骤可以被机器自动执行。算法可以用英文或者其他任何自然语言来描述，但是为了更加精确，往往会用伪代码进行描述。之所以称为“伪代码”，是因为它尚未细化到足以在计算机上运行的程度，与用编程语言编写的代码不大一样。

当今世界有越来越多的人以计算机编程为业。为了学习编程，我们首先要通过简单的例子学习基本的编程结构，例如赋值语句和控制循环之类，而编程习题往往涉及将算法的伪代码转译为用所学编程语言编写的代码。程序员同样能从求解谜题所需的分析技能中获益。无论是将规格说明转换为编程结构，还是定位早期代码中的错误（也就是调试过程），这些分析技能都不可或缺。

我在 MIT 教大一和大二学生编程有几十年了，我越来越清楚学生对实际应用的兴趣更加强烈。很少有人愿意为了编程而编程。谜题（puzzle）就是一种最棒的实际应用，它的优势在于易于描述和引人注目。后者如今尤为重要，因为讲课者必须和 Snapchat、Facebook 和 Instagram 争夺注意力。正如我的前辈们一样，我也发现让学生犯困的最佳方法就是介绍讨厌的编程语法或语义，只要几分钟就够了。

本书是我对编程教学的一次尝试，在算法谜题的趣味性与计算机编程的实用性之间架起一座桥梁。这里假定读者对编程的基本概念已有起码的了解，这些概念可以通过学习高中阶段的计算机科学入门课或大学预修（Advanced Placement, AP）课获得，也可以通过学习 MITx/edX 6.0001x 之类的课程获得。

本书中每个谜题的开始都会介绍一道谜题，其中不少谜题都脍炙人口，以各种变体形式在一些出版物和网站上出现过。在经历一两次失败的解谜尝试之后，突然灵光一闪，一种搜索策略、一个数据结构、一个数学公式跃然而出，答案就这么自行现身了。有时候会对谜题给出明显“暴力”的解法，本书会先解释相关的算法与代码，再将其解释为“失败”，然后再“捧出真经”，引出更加优雅和高效的解法。

谜题的解法正是需要编写的代码的规格说明书。读者要先了解代码要做的事情，

然后再看代码。我坚信这是一种很强大的教学理念，因为这把对代码功能的理解与编程语言语法和语义的理解分离开来。对于理解代码所需的语法和语义，将本着“现学现用”的原则进行介绍。

由谜题的物理世界到程序的计算机世界虽然很有趣，但并不总是一帆风顺。在某些情况下，必须假定某些操作在计算机世界中效率低下，因为其在物理世界中就是如此。本书会尽量减少这种情况的出现，但是仍无法完全避免。相信这不会对学习造成困扰，而且在极少出现的几处地方，书中都会明确指出。

读者可以采取多种方式来阅读和使用本书。如果仅对谜题本身及其答案感兴趣，完全可以在想出自己的解法或阅读本书给出的解法后即刻停止。但我不希望读者就此止步，因为讲解如何得出解法并转成可执行代码是写作本书的主要目的。读完整个谜题将对编写实用的程序所需的要素有很好的感知，实用的程序能供任何人自行运行和使用。我尽力确保对 Python 语法和语义的介绍能满足解谜代码的需要，但如果读者对 Python 语法、语义和库等有任何疑问，则 Python 的官方网站上有极佳的学习资源，edX/MITx 6.0001x 课程也有对 Python 编程的很好介绍。

如果读者能在自己的计算机上安装和运行 Python，就会从本书收获更多东西。这可以通过访问 Python 的官方网站来实现。书中所有谜题解法的代码都可以从 MIT 出版社网站下载。这些代码已在 Python 2.7 及以上版本（包括 3.x 版本）中测试过。当然，欢迎读者忽略这些可下载的代码，而编写自己的解题代码。强烈建议读者采用与书中示例不同的输入运行一下从网站下载或自行编写的程序。尽管我确实已尽力去除程序中的 bug，但仍不能保证代码没有错误。注意，加入对输入的检查会让代码显得零乱，因此，书中给出的代码假定输入符合谜题的描述，如果收到出乎意料的输入就不一定会有预期的表现了。为了加深对编程的理解，有一种最好的方式就是不断改进每一个谜题的代码，严格地检查格式错误的输入。

在每个谜题的结尾都会有几道编程的习题。这些习题的难度与所需编写的代码量各有不同。做完每个谜题的习题，将有助于充分掌握本书的内容。读者必须对谜题相关的代码有足够的理解，然后才有能力对其做出修改或改进其功能。书中的一部分习题包含了对错误输入的检查。标为“难题”的习题需要相当大的代码量，或要对已给出的谜题代码进行重构。其中的一部分难题可视为已介绍谜题的高级版本。本书没有提供习题和谜题的答案，教学人员可到 MIT 出版社网站的本书页面去获取。

我始终坚信应通过实践来学习，如果你能成功地自行完成所有习题，你将顺利成为一名计算机科学家！祝你好运。

致 谢

我在我的母校加州大学伯克利分校休假期间写了这本书。感谢 Robert Brayton 教授让我借用他在 Cory Hall 的办公室，在那里我完成了本书的大部分内容。感谢我在伯克利的房东 Sanjit Seshia、Kurt Keutzer 和 Dawn Song 让我享受了一个愉快且富有成果的假期。

我第一次教软件工程是与 Daniel Jackson 一起，这门课程使用 Java 作为编程语言。Daniel 对编程语言和软件工程的看法对我影响深远。我们在 JavaScript 加速器研讨会的合作中，为了更好地讲解基本的编程概念，出了一些谜题，例如通过不同的方式计算邮费。

我第一次教授计算机科学与编程导论是与 John Guttag 一起，这门课程主要面向非计算机科学专业的学生。John 的教学富有热情、非常有感染力，让我也喜欢上了使用 Python 教授编程导论。在本书中，我借鉴了不止一道他在教学中使用过的例题，例如通过二分搜索求平方根。

本书使用的一些谜题源于我与 Adam Chlipala 和 Ilia Lebedev 在 2015 年到 2016 年间合作开发的一门课程——编程基础 (Fundamentals of Programming)。在课程的课件与作业中，大家贡献了从作为草稿的演示代码到“生产级别”的可用代码。尤其感谢 Zhang Yuqing、Rotem Hemo 和 Jenny Ramseyer，他们分别编写了谜题 11 “请铺满庭院”、谜题 15 “统计零钱的组合方式”和谜题 21 “问题有价”的代码。大约有 400 名学生选修了 2017 年春季的编程基础课程，与 Duane Boning、Adam Hartz 和 Chris Terman 一起教这门课是一段非常愉快的经历。

谜题 3 “拥有（需要一点校准的）读心术”曾是计算机科学的数学 (Mathematics for Computer Science) 这门课的经典题目，我并不清楚它来自谁的发明，但感谢 Nancy Lynch 向我推荐了这道题目，并且感谢她在我为 15 年前的第一堂课伤神不已时扮演了“魔术助手”。这道谜题的描述来自三位讲师 Eric Lehman、Tom Leighton 和 Albert Meyer 的课堂笔记。

Kaveri Nadhamuni 也在谜题 6 “寻找假币”、谜题 8 “猜猜谁不来吃晚餐”、谜题 9 “美国达人秀”和谜题 16 “贪心是好事”上给了我帮助。Eleanor Boyd 在“代码女孩

计划”中测试了“寻找假币”这道谜题，并给出了很有价值的反馈。

Ron Rivest 对本书中的许多谜题提出了优化和总结，这些谜题包括谜题 1 “保持一致”、谜题 2 “参加派对的最佳时间”、谜题 5 “请打碎水晶”、谜题 8 “猜猜谁不来吃晚餐”、谜题 17 “字母也疯狂”和谜题 18 “充分利用记忆”。

Billy Moses 仔细阅读了本书，并给出了许多建议。

与同事 Costis Daskalakis、Erik Demaine、Manolis Kellis、Charles Leiserson、Nancy Lynch、Vinod Vaidyanathan、Piotr Indyk、Ron Rivest 和 Ronitt Rubinfeld 一起教授算法导论 (Introduction to Algorithms)、算法设计与分析 (Design and Analysis of Algorithms) 这两门课程的经历，使我的算法能力得到了提高。与同事 Saman Amarasinghe、Adam Chlipala、Daniel Jackson、John Guttag 和 Martin Rinard 一起教授软件课程的经历，增强了我对软件工程和编程语言的了解。感谢能与这些优秀的同事共事。

感谢 Victor Costan 和 Staphany Park 为编程基础 (Fundamentals of Programming) 与算法导论 (Introduction to Algorithms) 这两门课开发的课程打分系统，这套系统非常美妙，使我能将更多时间投入课程的内容本身。

MIT 出版社将本书的早期版本寄给了 3 位匿名的审校者。感谢他们仔细阅读书稿，并为改进本书提出了大量绝佳的建议。当然，也感谢他们的出版提议。希望他们读到本书时，能为我对他们有价值反馈的重视而感到高兴。

MIT 一直都鼓励教师在学科建设之外，也要重视实验室管理和领导力。我要感谢 Duane Boning、Anantha Chandrakasan、Agnes Chow、Bill Freeman、Denny Freeman、Eric Grimson、John Guttag、Harry Lee、Barbara Liskov、Silvio Micali、Rob Miller、Daniela Rus、Chris Terman、George Verghese 和 Victor Zue 在这些年里对我的支持。

Marie Lufkin Lee、Christine Savage、Brian Buckley 和 Mikala Guyton 为本书的审校、编辑、出版劳力颇多，感谢他们的工作。

我的太太 Lochan 给本书起了名字，我的女儿 Sheela 和 Lalita 是本书的首批“买家”，也帮助了本书的塑造。谨以本书献给她们 3 个人。

资源与支持

本书由异步社区出品，社区（<https://www.epubit.com/>）为您提供相关资源和后续服务。

配套资源

本书提供源代码下载，要获得此配套资源，请在异步社区本书页面中点击 **配套资源**，跳转到下载界面，按提示进行操作即可。注意：为保证购书读者的权益，该操作会给出相关提示，要求输入提取码进行验证。

提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，点击“提交勘误”，输入勘误信息，点击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的100积分。积分可用于在异步社区兑换优惠券、样书或奖品。

详细编辑

写书评

提交勘误

页码: 页码位置 (行/列): 勘误内容:

B I U 加粗 斜体 下划线 删除线

评论数: 0

提交

扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。



与我们联系

我们的联系邮箱是 contact@epubit.com.cn。

如果您对本书有任何疑问或建议，请您发邮件给我们，并请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 www.epubit.com/selfpublish/submission 即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为作译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 <https://www.epubit.com>。

“异步图书”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术等。



异步社区



微信服务号

目 录

谜题 1 保持一致	1
1.1 寻找想法相同的连续人员	2
1.2 字符串、列表和元组	3
1.3 从算法到代码	4
1.4 代码优化	7
1.5 列表创建与修改	7
1.6 作用域	8
1.7 算法优化	9
1.8 单遍算法	9
1.9 应用	10
1.10 习题	11
谜题 2 参加派对的最佳时间	13
2.1 反复检查时间	14
2.2 聪明地检查时间	16
2.3 有序地表示	20
2.4 习题	20
谜题 3 拥有（需要一点校准的）读心术	22
3.1 编程完成助手的工作	24
3.2 编程完成魔术师的任务	28
3.3 独自掌握技巧	29

3.4	信息编码	31
3.5	4 张牌的魔术戏法	31
3.6	习题	32
谜题 4	让皇后保持分离	34
4.1	系统地搜索	36
4.2	用二维列表（数组）表示棋盘	38
4.3	用一维列表（数组）表示棋盘	41
4.4	迭代枚举	45
4.5	习题	46
谜题 5	请打碎水晶	47
5.1	两颗球的高效搜索	48
5.2	d 颗球的高效搜索	49
5.3	对两颗球减少抛球次数	53
5.4	习题	54
谜题 6	寻找假币	55
6.1	分治	55
6.2	递归分治	57
6.3	三进制表示	60
6.4	称量谜题一个流行的变体	61
6.5	习题	61
谜题 7	跳到平方根	62
7.1	迭代查找	62
7.2	折半查找	65
7.3	二分搜索	67

7.4 三分搜索	69
7.5 习题	69
谜题 8 猜猜谁不来吃晚餐	71
8.1 第一次尝试	72
8.2 始终寻找最大选择	73
8.3 生成所有组合	74
8.4 移除不友好的组合	76
8.5 选择最大组合	76
8.6 优化内存使用	77
8.7 应用	78
8.8 习题	79
谜题 9 美国达人秀	81
9.1 每次生成并测试一个组合	83
9.2 确定缺少一门绝活的组合	84
9.3 应用	85
9.4 习题	86
谜题 10 多皇后	88
10.1 递归求取最大公约数	88
10.2 递归获取斐波那契数列	89
10.3 递归求解 N 皇后问题	91
10.4 递归的应用	94
10.5 习题	96
谜题 11 请满铺庭院	98
11.1 归并排序	99

11.2	归并排序的执行与分析	101
11.3	基线条件即 2×2 庭院	102
11.4	递归步骤	103
11.5	列表推导式的基础知识	107
11.6	美观打印	107
11.7	另一个满铺谜题	109
11.8	习题	109
谜题 12 汉诺塔		111
12.1	汉诺塔的递归解决方案	112
12.2	相邻汉诺塔的递归解决方案	114
12.3	与格雷码的关系	117
12.4	习题	118
谜题 13 没条理的工匠		121
13.1	分治时的围绕基准点分拆	122
13.2	与排序算法的关系	123
13.3	原地划分	126
13.4	排序也疯狂	129
13.5	习题	129
谜题 14 再也不玩数独了		131
14.1	递归式数独求解	132
14.2	递归搜索过程中的推理	136
14.3	数独谜题的难度	140
14.4	习题	141
谜题 15 统计零钱的组合方式		143
15.1	钞票的递归选取	143

15.2 消除重复	145
15.3 用最少的钞票支付	147
15.4 习题	148
谜题 16 贪心是好事	150
16.1 贪心算法	151
16.2 最短历时规则	151
16.3 最早开始时间规则	151
16.4 最少冲突规则	152
16.5 最早结束时间规则	152
16.6 贪心算法何时有效	157
16.7 习题	158
谜题 17 字母也疯狂	160
17.1 每次找到一组变位词	160
17.2 通过排序对变位词进行分组	162
17.3 通过散列操作对变位词进行分组	164
17.4 字典	165
17.5 用字典对变位词进行分组	167
17.6 散列表	169
17.7 习题	171
谜题 18 充分利用记忆	173
18.1 递归解决方案	173
18.2 回溯硬币的选择过程	175
18.3 memoization 技术	178
18.4 避免使用异常	179
18.5 动态规划	180
18.6 习题	180

谜题 19 要记得周末	184
19.1 找到分区	185
19.2 二分图的判别	187
19.3 图的表示	189
19.4 图的着色	192
19.5 习题	193
谜题 20 六度分隔	195
20.1 广度优先搜索	197
20.2 集合	198
20.3 在广度优先搜索中使用集合	199
20.4 历史	202
20.5 习题	203
谜题 21 问题有价	205
21.1 用字典构造二叉查找树	207
21.2 字典形式下的二叉查找树操作	209
21.3 面向对象风格的二叉查找树	212
21.4 回到谜题：算法	216
21.5 解决谜题的代码	218
21.6 多种数据结构的对比	222
21.7 习题	222

谜题 1

保持一致

本谜题涵盖的编程结构和算法范型：列表、元组、函数、控制流程（包含 `if` 语句和 `for` 循环）和 `print` 语句。

假设有一大群人排队等待观看棒球比赛。他们都是主场球迷，每个人都戴着队帽。但不是所有人都用同一种戴法，有些人正着戴，有些人反着戴。

人们对正戴和反戴的定义各不相同，但你认为图 1-1 中左边的帽子是正戴的，右边的帽子是反戴的。

假定你是门卫，只能在全组球迷帽子戴法一致时才能让他们进入球场，要么全部正着戴，要么全部反着戴。因为每个人对帽子正反的理解不同，所以不能对他们说把帽子正着戴或反着戴。只能告诉他们转一下帽子。好消息是，球迷每个人都知道自己在队伍中的位置，第一个人的位置是 0，最后一个人的位置是 $n-1$ 。这可以像下面这样表述。

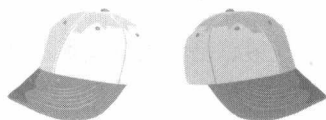


图 1-1

位置 i 的人请转一下帽子。

位置 i 到 j （包含 j ）的人请转一下帽子。

不过你想要尽量减少喊出要求的次数。举个例子，如图 1-2 所示。

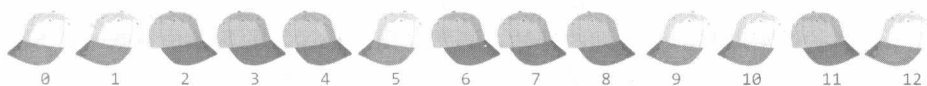


图 1-2

这是一个有 13 个人的队伍，位置从 0 到 12。因为有 6 个人的帽子是正戴的，应

该喊出 6 次命令。例如：

位置 0 的人请转一下帽子。

并对位置 1、5、9、10 和 12 的人重复相同的命令。采用第二种方式的命令可以减少说话次数，只需要喊出下面 4 次命令，就会让所有人的帽子都反戴。

位置 0 到 1 的人请转一下帽子。

位置 5 的人请转一下帽子。

位置 9 到 10 的人请转一下帽子。

位置 12 的人请转一下帽子。

但对这个例子而言，还能做得更好一些。如果喊出以下命令，会让所有人的帽子都变成正戴。

位置 2 到 4 的人请转一下帽子。

位置 6 到 8 的人请转一下帽子。

位置 11 的人请转一下帽子。

怎样才能让生成的命令数最少呢？难度更大的问题是，能否第一次沿着队伍走一遍就想出生成命令的方案？

在继续往下阅读以前，请思考一下上述问题。

1.1 寻找想法相同的连续人员

假定有了对应于一组排队等待人员的帽子方向列表。可以计算出一个“区间”（interval）的列表，对应于各段戴帽方式相同的连续人群。区间可以用开始和结束位置来表示，如 $[a, b]$ ($a \leq b$)，这表示 a 和 b 之间所有位置的区间，包括 a 和 b 在内。

每个区间都标上是正戴区间还是反戴区间。所以一个区间有 3 个属性，即开始位置值、结束位置值、标明正戴或反戴的标记。

如何计算出区间的列表呢？有一个关键点就是，当看到帽子方向发生变化时，就是一个区间的结束和另一个区间的开始。当然，第一个区间的起始位置是 0。下面再给出一次例子，如图 1-3 所示。